

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-232115

(43)Date of publication of application : 27.08.1999

(51)Int.Cl.

G06F 9/445

G06F 13/00

G06F 15/16

(21)Application number : 10-296431

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 19.10.1998

(72)Inventor : MASSEY ANDREW LIAM
SYED SOHAIL

(30)Priority

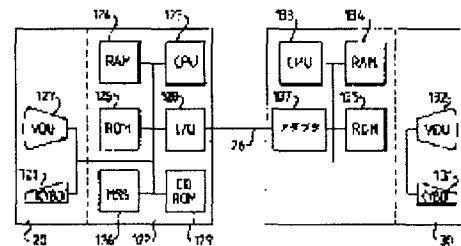
Priority number : 97 9724364 Priority date : 19.11.1997 Priority country : GB

(54) SOFTWARE MODULE TRANSFER METHOD AND DATA PROCESSING NETWORK AND SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To consider the activity of a user in a work station and to reduce the loss of data transmission time between a client system and a server by predicting the type of a software module based on a using history and automatically performing transfer based on the monitoring of a traffic level.

SOLUTION: A server system 20 transmits a first software module type to a connected client system 30. A control software predicts the type of the software module assumed to be requested by the same client based on a data gathering software or compiled history information. The control software decides whether or not LAN traffic is smaller than a certain threshold value, and when it is so, makes a push software push the predicted module to the client 30.



LEGAL STATUS

[Date of request for examination] 27.07.1999

[Date of sending the examiner's decision of rejection] 30.05.2000

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 2000-10577

[Date of requesting appeal against examiner's decision of rejection] 12.07.2000

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-232115

(43) 公開日 平成11年(1999) 8月27日

(51) Int.Cl. ⁸	識別記号	F I
G 0 6 F 9/445		G 0 6 F 9/06
13/00	3 5 1	13/00
15/16	6 2 0	15/16
		4 2 0 C
		3 5 1 H
		6 2 0 B

審査請求 未請求 請求項の数16 OL (全 10 頁)

(21) 出願番号 特願平10-296431

(22) 出願日 平成10年(1998)10月19日

(31) 優先権主張番号 9 7 2 4 3 6 4. 6

(32) 優先日 1997年11月19日

(33) 優先権主張国 イギリス (G B)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72) 発明者 アンドリュー・リーム・マッセイ

イギリス ケー・エー30 9ビー・ワイ
アーシュレー ラーガス パーンサイド・ロード 61

(74) 代理人 弁理士 坂口 博 (外1名)

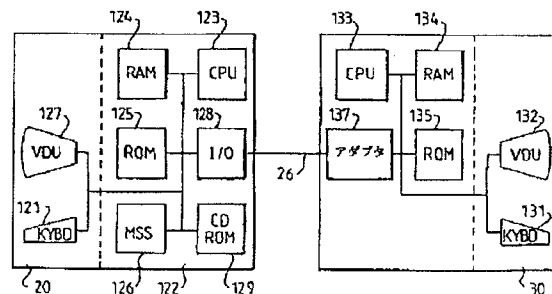
最終頁に続く

(54) 【発明の名称】 ソフトウェア・モジュール転送方法、データ処理ネットワークおよびシステム

(57) 【要約】

【課題】 データ処理ネットワーク中でソフトウェア・モジュール (例えば j a v a アプレットなど動的にロードできるコード) をクライアント・データ処理システム上で実行するためにホスト・データ処理システムからブリエンプティブに転送する方法およびシステムを提供すること。

【解決手段】 ソフトウェア・モジュールは、それぞれクライアント・システムのところで異なるサービス (例えば電子メール、テキスト・エディタなど) を提供する異なる多数のタイプである。ホスト・システムは、クライアントに転送される第1のソフトウェア・モジュールのタイプを決定し、この決定に基づいて、次にクライアントのところで必要とされるであろうソフトウェア・モジュールのタイプに関して予測を行う。次いで、予測したソフトウェア・モジュールが、クライアントから同じものを要求される前にクライアント・システムに転送される。



【特許請求の範囲】

【請求項 1】データ処理ネットワーク中でソフトウェア・モジュールをターゲット・データ処理システム上で実行するためにホスト・データ処理システムからプリエンブディブに転送する方法であって、それぞれ前記ターゲット・システムのところと異なるサービスを提供する複数のタイプのソフトウェア・モジュールがあり、前記ホスト・システムのところと、ターゲット・システムに転送される第 1 のソフトウェア・モジュールのタイプを決定するステップと、

前記第 1 のモジュールのタイプの決定に基づいて、次に前記ターゲット・システムのところと必要とされるであろう第 2 のソフトウェア・モジュールのタイプを予測するステップと、
前記第 2 のタイプのモジュールを前記ターゲット・システムに自動的に転送するステップとを含む方法。

【請求項 2】前記サーバのところと、各ターゲット・システムにおけるソフトウェア・モジュール使用の履歴をコンパイルする他のステップを含み、前記予測ステップが各ターゲット・システムの前記コンパイルされた履歴

に基づく請求項 1 に記載の方法。
【請求項 3】前記ネットワーク上のトラフィックのレベルを監視し、かつ前記トラフィックがあるしきい値よりも小さいときだけ前記第 2 のソフトウェア・モジュールを自動的に転送する他のステップを含む請求項 1 または請求項 2 に記載の方法。

【請求項 4】前記第 1 のソフトウェア・モジュールが前記ターゲット・システムからの要求に回答して、前記ホスト・システムから転送される請求項 1 ないし 3 のいずれか一項に記載の方法。

【請求項 5】前記ソフトウェア・モジュールが動的にロードできるソフトウェア・モジュールである請求項 1 ないし 4 のいずれか一項に記載の方法。

【請求項 6】前記動的にロードできるソフトウェア・モジュールが、前記ターゲット・システム上で `java` 使用可能ブラウザまたは `java` 仮想マシンを介して実行するための `java` アプレットである請求項 5 に記載の方法。

【請求項 7】前記ネットワークがローカル・エリア・ネットワークである請求項 1 ないし 6 のいずれか一項に記載の方法。

【請求項 8】前記ネットワークがインターネットまたはイントラネットである請求項 1 ないし 6 のいずれか一項に記載の方法。

【請求項 9】前記第 1 のソフトウェア・モジュールが前記ターゲット・システムのところとテキスト編集サービスを提供する請求項 1 ないし 8 のいずれか一項に記載の方法。

【請求項 10】前記第 2 のソフトウェア・モジュールが前記ターゲット・システムのところと電子メール通信サ

ービスを提供する請求項 9 に記載の方法。

【請求項 11】複数のターゲット・データ処理システムと通信するために接続されたサーバ・データ処理システムを含むデータ処理ネットワークであって、前記サーバ・データ処理システムのところにあるか、またはそれに関連し、それぞれターゲット・システムのところと異なるサービスを提供する複数のタイプのソフトウェア・モジュールを前記複数のターゲット・システムにその上で実行するために転送する手段と、

10 前記ターゲット・システムに送信された前記ソフトウェア・モジュールの識別を監視する手段と、

第 1 のタイプのソフトウェア・モジュールがターゲット・システムと通信していたという決定に回答して、第 2 のタイプのソフトウェア・モジュールが次に前記ターゲット・システムのところと必要とされるであろうと予測し、かつ前記転送手段に前記第 2 のタイプのソフトウェア・モジュールを前記ターゲット・システムにプリエンブディブに転送させる手段と、

前記ターゲット・システムのところと、前記ホスト・システムからの前記ソフトウェア・モジュールを受信する手段とを含むデータ処理ネットワーク。

【請求項 12】各前記ターゲット・システムにおけるソフトウェア・モジュール使用の履歴をコンパイルするデータ収集手段をさらに含み、前記予測手段が前記予測を各ターゲット・システムの前記履歴を基にする請求項 11 に記載のデータ処理ネットワーク。

【請求項 13】前記ネットワーク上のトラフィックのレベルを監視し、かつ前記トラフィックがあるしきい値よりも小さいときだけ前記第 2 のソフトウェア・モジュールを自動的に転送する手段をさらに含む請求項 11 または請求項 12 に記載のデータ処理ネットワーク。

【請求項 14】各ソフトウェア・モジュールのタイプが動的にロードできるソフトウェア・モジュールである請求項 11 ないし請求項 13 のいずれかに記載のデータ処理ネットワーク。

【請求項 15】前記動的にロードできるソフトウェア・モジュールが `java` アプレットであり、各ターゲット・システムが `java` 使用可能ブラウザまたは `java` 仮想マシンを含む請求項 14 に記載のデータ処理ネットワーク。

【請求項 16】複数のクライアント・データ処理システムと通信する手段と、

それぞれ前記ターゲット・システムのところと異なるサービスを提供する複数のタイプのソフトウェア・モジュールを前記複数のクライアント・システムにその上で実行するために転送する手段と、

前記ターゲット・システムに送信された前記ソフトウェア・モジュールの識別を監視する手段と、

第 1 のタイプのソフトウェア・モジュールがターゲット・システムと通信していたという決定に回答して、第 2

のタイプのソフトウェア・モジュールが次に前記ターゲット・システムのところで必要とされるであろうと予測し、かつ前記転送手段に前記第 2 のタイプのソフトウェア・モジュールを前記ターゲット・システムにプリエンプティブに転送させる手段とを含むサーバ・データ処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ処理ネットワーク中でソフトウェアをホスト・システムからターゲット・システムにプリエンプティブに転送する方法に関する。

【0002】

【従来の技術】代表的なネットワーク環境中では、複数のクライアント・データ処理システムが 1 つまたは複数のサーバ・システムに接続される。第 1 の一般的な構成では、各クライアント・システムは、クライアント中のハード・ファイル上に格納されるオペレーティング・システムおよび任意選択で他のソフトウェアを含む。サーバ・システムに関連する記憶装置上に保持される他のアプリケーション・ソフトウェア、例えばワード・プロセッシング、データベース・ソフトウェアは、クライアント・システムによって必要なときにアクセスされる。

【0003】代替ネットワーク構成では、クライアント・システムは、ほとんどまたはまったく大容量記憶能力を有せず、この場合、オペレーティング・システムおよびユーザが必要とする他のソフトウェアは、クライアント・システムによって必要なときにサーバからダウンロードされ、揮発性メモリ中に記憶される。

【0004】現存のネットワークでは、ソフトウェアおよび他のデータは、一般に、クライアント・システムの要求によりサーバから転送される。これは、クライアント・ユーザ側の特定のアクションか、またはクライアント上での処理の実行中に自動的に行われる。サーバ・システムが使用中のとき、ソフトウェアに対する要求は、待ち行列中でそれらの順番を待っており、それにより大きい遅延が生じることがある。さらに、待ち行列がまったくなく、かつソフトウェアに対する要求がサーバのところで受信された直後に処理された場合でも、それにもかかわらず、要求されたソフトウェアを製造し、要求されたソフトウェアを接続リンクを介してクライアントに伝達する際に必要な時間の損失と共に要求側クライアントが要求をサーバに送信する必要によって生じる時間の損失がある。ネットワーク・トラフィックが多いとき、生じる全遅延は大きくなり、クライアント・システムのユーザの活動に悪影響を及ぼすことがある。

【0005】米国特許出願第 5 0 2 9 1 0 4 号には、ホスト・システムが接続されたワークステーションのところでいくつかのタイプのデータ・オブジェクトの必要を

予測する技法が記載されている。ホストは、ホストのところで受信されるデータ・オブジェクト（例えば、ワークステーション・ユーザにとって重要な電子メール）にตอบสนองして、ワークステーションが各オブジェクトを使用する可能性を決定するヒューリスティック論理回路を含む。ワークステーションのところで使用される可能性のあるデータ・オブジェクトは、ワークステーションによって要求が行われる前にホストによって任意にワークステーションに送信される。

【0006】したがってこの従来技術では、ホスト・システムは、データ・オブジェクトがワークステーション・ユーザにとって有用または重要であるという決定に基づいてデータ・オブジェクトをワークステーションにプリエンプティブに送信する。この決定は、ホスト・システムにおけるデータ・オブジェクトの受信に基づいており、ワークステーションにおけるユーザの活動を考慮しない。

【0007】

【発明が解決しようとする課題】本発明の第 1 の態様によれば、データ処理ネットワーク中でソフトウェア・モジュールをターゲット・データ処理システム上で実行するためにホスト・データ処理システムからプリエンプティブに転送する方法であって、それぞれターゲット・システムのところで異なるサービスを提供する複数のタイプのソフトウェア・モジュールがあり、ホスト・システムのところで、ターゲット・システムに転送される第 1 のソフトウェア・モジュールのタイプを決定するステップと、第 1 のモジュールのタイプの決定に基づいて、ターゲット・システムのところで次に必要とされるであろう第 2 のソフトウェア・モジュールのタイプを予測するステップと、第 2 のタイプのモジュールをターゲット・システムに自動的に転送するステップとを含む方法が提供される。

【0008】本発明の第 2 の態様によれば、複数のクライアント・データ処理システムと通信する手段と、それぞれターゲット・システムのところで異なるサービスを提供する複数のタイプのソフトウェア・モジュールを複数のクライアント・システムにその上で実行するために転送する手段と、ターゲット・システムに送信されたソフトウェア・モジュールのタイプを監視する手段と、第 1 のタイプのソフトウェア・モジュールがターゲット・システムと通信していたという決定にตอบสนองして、第 2 のタイプのソフトウェア・モジュールが次にターゲット・システムのところで必要とされるであろうと予測し、かつ転送手段に第 2 のタイプのソフトウェア・モジュールをターゲット・システムにプリエンプティブに転送させる手段とを含むサーバ・データ処理システムが提供される。

【0009】

【課題を解決するための手段】したがって本発明によれ

ば、クライアント・システムに転送されたソフトウェア・モジュールのタイプに基づいて、次にクライアントが必要とするであろうソフトウェア・モジュールのタイプに関して予測を行う方法およびシステムが提供される。好ましい構成では、予測は、各クライアントに関連するソフトウェア・モジュール使用の履歴に基づいており、この履歴は、一定の時間期間にわたってサーバによってコンパイルされる。

【0010】本発明の好ましい実施形態では、ネットワーク上のトラフィックのレベルを監視し、測定したトラフィックがあるしきい値よりも小さいときだけ予測したモジュールを自動的に転送する。

【0011】1つの好ましい構成では、ソフトウェア・モジュールは、クライアント・システム上でjava使用可能ブラウザまたはjava仮想マシンを介して実行するために動的にロードできる（例えば、Javaアプレット）。

【0012】本発明の他の態様によれば、コンピュータ使用可能媒体中に組み込まれ、データ処理ネットワーク中でソフトウェア・モジュールをターゲット・データ処理システム上で実行するためにホスト・データ処理システムからプリエンティブに転送するコンピュータ読取り可能プログラム・コード手段を有するコンピュータ使用可能媒体を含むコンピュータ・プログラム製品であって、それぞれターゲット・システムのところで異なるサービスを提供する複数のタイプのソフトウェア・モジュールがあり、コンピュータ読取り可能プログラム・コード手段が、ホスト・システムのところで、ターゲット・システムに転送される第1のソフトウェア・モジュールのタイプを決定するコンピュータ読取り可能プログラム・コード手段と、第1のモジュールのタイプの決定に基づいて、次にターゲット・システムのところで必要とされるであろう第2のソフトウェア・モジュールのタイプを予測するコンピュータ読取り可能プログラム・コード手段と、第2のタイプのモジュールをターゲット・システムに自動的に転送するコンピュータ読取り可能プログラム・コード手段とを含むコンピュータ・プログラム製品が提供される。

【0013】次に、本発明の好ましい実施形態について、添付の図面を参照しながら例としてのみ説明する。

【0014】

【発明の実施の形態】まず第一に図1を参照すると、本発明の好ましい実施形態が実施されるローカル・エリア・ネットワーク（LAN）10が概略的な形で示されている。図1のネットワークは、イーサネットLANとして構成されるものとして示されているが、代わりにトークン・リングまたは他の適切なトポロジに従って構成することもできる。以下で説明するように、本発明はまた、インターネット・ネットワークまたはイントラネット・ネットワークのコンテキストにおいても使用でき

る。図1において、ネットワークは、複数のクライアント・コンピュータ・システム30、32、34、36と通信するためにリンク26によって接続されたサーバ・コンピュータ・システム20（この実施形態では、IBM PC 730コンピュータ・システム）を含む。クライアント・コンピュータ・システムは、インテルX86ファミリーのマイクロプロセッサをベースにしたパーソナル・コンピュータまたは他の形のコンピュータ・システムである。以下で説明するように、クライアントはあるいはJava仮想マシンまたはJava使用可能ブラウザを含む「ネットワーク・コンピュータ」（例えばIBMネットワーク・ステーション）である。各クライアント・システムは、リンク26を介してサーバ・コンピュータとの通信を実施するLANアダプタ・カードまたはネットワーク・インタフェース・カード（NIC）38、40、42、44を含む。

【0015】サーバとクライアントとの間の通信を含むネットワーク上の資源の制御は、ネットワーク・オペレーティング・システム（NOS）、例えば、サーバ・コンピュータ・システムの主プロセッサ上で実行する「サーバ」構成要素と、各クライアント・コンピュータ・システムの主プロセッサ上で実行する対応する「リクエスト」構成要素とを有するNovell社のNetwareを使用して実施される。他の適切なネットワーク・オペレーティング・システムには、どちらもIBM社のOS/2 LANサーバおよびOS/2 WARPサーバがある。この実施形態では、Internetwork Packet Exchange（IPX）通信プロトコルがNetwareオペレーティング・システムに関連して使用される。他の実施形態では、TCP/IPが好ましいプロトコルである。

【0016】図2は、通信リンク26を介してサーバ・コンピュータ・システム20をクライアント・システム30に接続する方法を示す簡略化されたブロック図である。クライアント・システムは、システム・バスによってシステム・メモリ（RAM）134およびPOSTコードを含むシステムBIOSがその中に格納された不揮発性メモリ（ROM）135に接続された主CPU133の形をした制御論理回路の制御下で動作するキーボード131およびディスプレイ132を含む。クライアント・システムはさらに、この実施形態ではイーサネット・カードであるが、代替実施形態ではトークン・リング・アダプタ・カードであるネットワーク・アダプタ・カード137を含む。このアダプタ・カードは、クライアントとサーバとの間で通信を実施する際に使用されるコードがその中に記憶されたROMの形をした不揮発性メモリを含む。

【0017】この実施形態のクライアント・システムは、例えば、磁気ディスク・ドライブ（ハード・ファイル）の形をした大容量記憶装置を備えていないことに留

意されたい。さらに、クライアント・ユーザがソフトウェアまたはデータをクライアント・システム中に導入するのを防ぎたい場合、クライアント・システムは、ディスク・ドライブ、CD-ROMドライブ、または同様のドライブを備えていないことが有利である。

【0018】図2のサーバ・コンピュータ・システムは、主CPU123、システムRAM124、システムROM125、および一般にRAID（独立したディスクの冗長アレイ）の形で構成された複数の磁気ディスク・ドライブの形をした大容量記憶装置126を含むシステム装置122に接続されたキーボード121を含む。サーバ・システムは、任意選択で、ディスプレイ127（ネットワーク管理者がサーバ・システムと直接対話する必要がある場合）およびディスク・ドライブ（図示せず）やCD-ROMドライブ129など他の記憶装置を含む。リンク26を介する通信は、アダプタ・カードの形をとる入出力論理回路128によって実施される。

【0019】サーバRAIDのディスク・ドライブ上には、オペレーティング・システム・ソフトウェアを含む異なる様々なソフトウェア、およびクライアント・システムにダウンロードし、またクライアント・システムが使用するための異なる様々なタイプのソフトウェア・モジュールが格納される。これらのソフトウェア・モジュールは、クライアント・システムのところで異なるサービスを提供するように設計された比較的小さいコードである。例えば、電子メール、テキスト編集、カレンダー、スプレッドシート、および他のサービスをクライアント・ユーザに提供するように設計されたモジュール・タイプがある。モジュールは、ユーザ（人間またはプロセス）の特定の要求によるか、または本発明によるプリエンティブな形でクライアント・システムに転送される。

【0020】ソフトウェア・モジュールは、クライアント・システム上に動的にロードできることが有利である。この場合、モジュールは、クライアントに転送されたとき、自動的にロードし、例えばウィンドウ内でクライアント・ユーザに表示される。モジュールはさらに、一度動作し、その後廃棄されるように設計することもできる。あるいは、モジュールは、後でクライアントが使用するためにキャッシュに格納することもできる。適切なソフトウェア・モジュールの一例は、Javaアプレットである（JavaはSun Microsystems社の登録商標である）。この場合、クライアント・システムは、Windows NTなど適切なオペレーティング・システム上で動作するJava使用可能ブラウザ（例えばNetscape Communications社から販売されているNetscape Navigator）またはJava仮想マシン（JVM）を含む。

【0021】図3を参照すると、本発明の好ましい実施形態の機能を提供するサーバ・システム上で実行するソフトウェア・コンポーネント200（この実施形態ではNetwareモジュール）のグループの相互関係が示されている。LAN解析ソフトウェア210（例えば、Novell社から販売されているLANAnalyzer）は、LAN上のトラフィックのレベルに関する情報をコントローラ205に提供するために従来の形でネットワーク上のトラフィックを監視する。多数の適切なLAN解析ソフトウェアが販売されている（例えば、Novell社から販売されているLANAnalyzer）。

【0022】データ収集ソフトウェア220は、様々なクライアント・システム上の様々なソフトウェア・モジュールの使用の履歴を作成するバックグラウンド・タスクとして実行するように設計される。この履歴は、特定のクライアントが要求したモジュールのモデル、モジュールが要求された時刻、ならびに異なるモジュール・タイプが使用される代表的なシーケンスを得るためにデータ収集タスクによって数日間または数週間にわたってコンパイルされる。

【0023】例えば、クライアントXは、既存の予定された会議の時刻を検査するためにかつ／またはさらに会議情報を追加するために常に営業日の始めにカレンダー・ソフトウェア・モジュールを要求する。この特定のユーザは、次いで前の営業日に関する金融情報または他の情報を更新するためにスプレッドシート・モジュールを常に要求する。別のクライアントYは、最初に電子メール・モジュールを要求し、その後カレンダー・モジュールを要求する。使用履歴は、LANが位置する組織内の個々のユーザの役割に応じてクライアント間で異なる可能性があることを理解されたい。

【0024】この履歴情報は、データ収集ソフトウェアによってコンパイルされ、本発明では、いずれかのクライアント・システムの将来の要件に関して予測を行うために使用される。したがって、クライアントXがカレンダー・モジュールを要求したとき、これはデータ収集タスクによって監視される。制御ソフトウェアは、データ収集ソフトウェアからの情報に基づいて、クライアント・システムXが次に使用する可能性のあるソフトウェア・モジュールのタイプを予測する予測アルゴリズム225を含む。この場合、履歴は、クライアントXが必ずカレンダー・モジュールの後でスプレッドシート・モジュールを要求することを示す。この場合、制御ソフトウェアは、プッシュ・タスク230にクライアントXのユーザからの要求の前にスプレッドシート・モジュールをクライアントXにプッシュさせる。好ましい構成では、次に予測されるモジュールの転送は、LANトラフィックが事前に定義されたかまたは事前に計算されたしきい値よりも小さいという（LAN解析タスクからのトラフィッ

ク情報に基づく) 決定によって決まる。この構成によれば、ネットワーク上の負荷のバランスを改善することができる。

【0025】代替構成では、様々なクライアントによるモジュールの使用の履歴を作成するデータ収集ソフトウェアはない。代わりに、予測は、ソフトウェア・モジュールのタイプ間の関連のリストに基づく。例えば、クライアントは、最初にテキスト・エディタを要求した場合、次に要求されるモジュールが電子メール通信モジュールであろうと予測することが妥当であろう。電子メール通信モジュールは、次いでクライアントがそのモジュールを要求する前にクライアントにプリエンブティブに転送される。モジュールの関連は、適切な形、例えばテーブルの形でサーバ・システム中に格納される。

【0026】予測モードにかかわらず、予測されたソフトウェア・モジュールは、クライアントによる予測された使用の直前に転送されることが理想的であるが、この場合、モジュールは、例えばサービスを表示するウィンドウがクライアントの画面上にポップアップするようにクライアントによって動的にロードされる。この理想的な構成を実現する能力は、データ収集ソフトウェアの精巧化、任意の時刻におけるネットワーク上の使用可能な帯域幅、ならびにクライアント・ユーザの予測可能性に依存する。動的にロードできるモジュールは、使用後に廃棄される(この場合モジュールはサーバによって再送信されなければならない)か、あるいは後続の呼出しを予測してキャッシュに格納される。

【0027】モジュールは、動的にロードできる必要はなく、むしろユーザによる後続の呼出しのためにクライアントのところのキャッシュ・メモリに転送される。任意の適切な構成を使用して、クライアント中のソフトウェア・モジュールのキャッシングを制御することができる。例えば、使用可能なキャッシュがいっぱいとき、ソフトウェア・モジュールがFIFO方式でページされるか、あるいは各モジュールがそれに優先度(例えば、テキスト・エディタがスプレッドシートよりも重要である)を関連付けており、モジュール優先度に従ってキャッシュ中のスペースを解放することができる。

【0028】次に図4を参照すると、本発明の好ましい方法に関連するステップの流れ図が示されている。この方法は、図示のように300のところから始まる。サーバ・システムは、ステップ302で、第1のソフトウェア・モジュール・タイプを接続されたクライアント・システムの1つに送信する。これは、クライアントの要求によるか、あるいはサーバによってプリエンブティブな形で実施される。制御ソフトウェアは、ステップ304で、データ収集ソフトウェアによってコンパイルされた履歴情報に基づいて、多分同じクライアントによって要求されるであろうソフトウェア・モジュールのタイプに関して予測を行う。ステップ306で、制御ソフトウェ

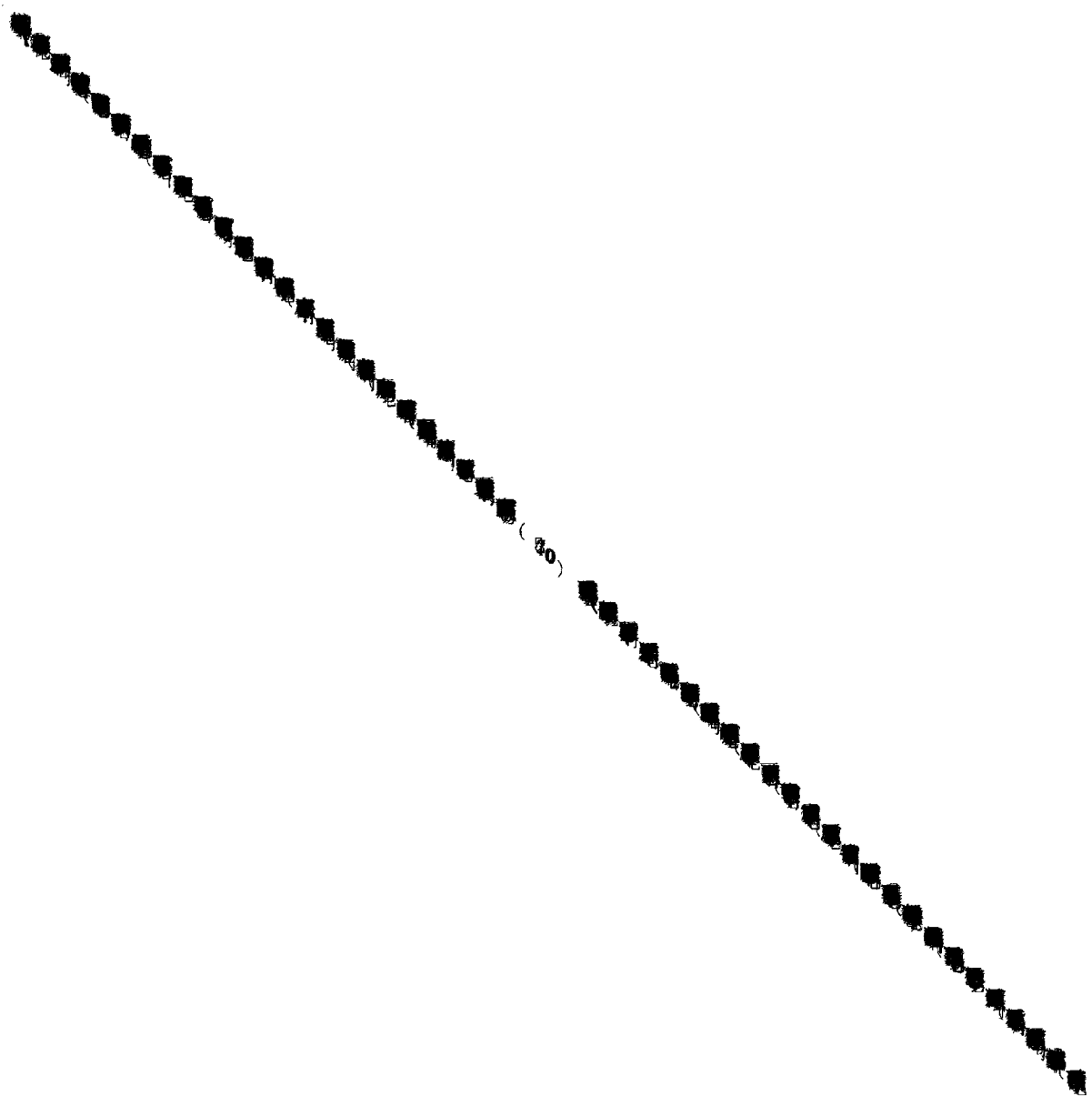
アは、LANトラフィックがあるしきい値よりも小さいかどうかに関する決定を行い、そうであれば、ステップ308で、プッシュ・ソフトウェアに予測したモジュールを問題のクライアントにプッシュさせる。このプロセスはステップ310で終了する。

【0029】既に示したように、特定のクライアントに関連する使用履歴は、そのクライアントのユーザに固有である。予測アルゴリズムの成功は、クライアント・システムが同じユーザによって使用されることに依存する。これは、一般にワークステーション・グループを含むオープン・オフィス環境中でもそうである。そのようなオフィス内の個人には、一般に、その上で作業すべき特定のシステムが割り当てられる。したがって、ユーザAが、例えば毎週月曜日にワークステーションWS10を使用する場合、データ収集ソフトウェアは、ソフトウェア・モジュール使用のパターンが月曜日から月曜日まで比較的一定していることを記す。したがって、サーバ・システムは、月曜日の朝にユーザAが到着する前に予測して適切なソフトウェア・モジュールをワークステーションWS1に供給する。同様に、サーバは、その日の残りの時間中にワークステーションWS1からの要求の前に予測して他のソフトウェア・モジュールを供給する。

【0030】何らかの理由でユーザBが月曜日の朝にワークステーションWS10を使用することを選択し、かつ通常WS10によって要求されないソフトウェア・モジュールに対する要求を行った場合、これはデータ収集ソフトウェアによって記される。これに基づいて、ソフトウェア・モジュールの通常の「月曜日」シーケンスを予測して供給する代わりに、制御ソフトウェアは、予測を行うのを中止するか、あるいは上述の形で純粋にソフトウェア・モジュール・タイプの関連に基づいてソフトウェア使用を予測する。

【0031】本発明はまた、インターネットまたはイントラネットのコンテキストにおいても使用できる。図5は、インターネットまたはイントラネット410を介してそのうちの3つがシステム412、414、416として示されている複数のクライアント・システムに接続されたホスト・システム400を含むそのような構成の概略図である。ホスト・システムは、図1のホストと同じ構成を有する。クライアント・システムは、Java使用可能ブラウザ420、またはホスト・システムから受け取ったJavaアプレット・ソフトウェア・モジュールを実行するJava仮想マシンを含む。各クライアントは、任意選択でソフトウェア・モジュールを一時的に記憶するキャッシュ422を含む。ホスト・システムは、クライアント・システムの個々のIPアドレスを参照して、モジュール使用の履歴をコンパイルし、維持することができる。

【0032】以上、好ましい実施形態について、大容量



13

シンを含む上記 (1 4) に記載のデータ処理ネットワーク。

(1 6) 複数のクライアント・データ処理システムと通信する手段と、それぞれ前記ターゲット・システムのところで異なるサービスを提供する複数のタイプのソフトウェア・モジュールを前記複数のクライアント・システムにその上で実行するために転送する手段と、前記ターゲット・システムに送信された前記ソフトウェア・モジュールの識別を監視する手段と、第 1 のタイプのソフトウェア・モジュールがターゲット・システムと通信して 10 いたという決定に応答して、第 2 のタイプのソフトウェア・モジュールが次に前記ターゲット・システムのところで必要とされるであろうと予測し、かつ前記転送手段に前記第 2 のタイプのソフトウェア・モジュールを前記ターゲット・システムにプリエンプティブに転送させる手段とを含むサーバ・データ処理システム。

【図面の簡単な説明】

【図 1】複数のクライアント・データ処理システムに接続されたホスト・システムを含むデータ処理ネットワークの概略図である。

【図 2】図 1 のクライアント・システムおよびホスト・システムの主要な構成要素の概略図である。

【図 3】本発明の好ましい実施形態において使用されるホスト・システムの要素間の対話を示す概略図である。

【図 4】本発明の好ましい一実施形態のステップを示す流れ図である。

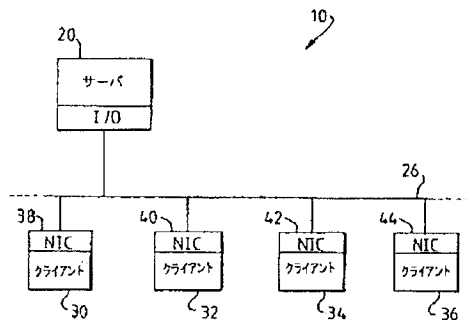
【図 5】本発明がそれと共に使用される代替ネットワーク構成の概略図である。

【符号の説明】

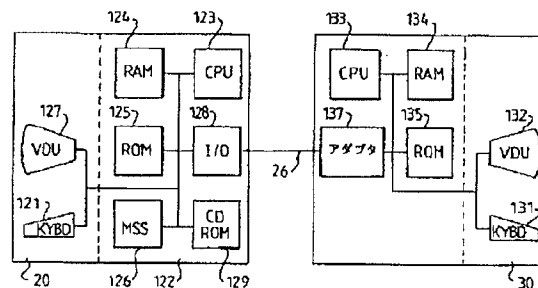
1 0 ローカル・エリア・ネットワーク (LAN)
2 0 サーバ・コンピュータ・システム
2 6 通信リンク
3 0 クライアント・コンピュータ・システム
3 2 クライアント・コンピュータ・システム

3 4 クライアント・コンピュータ・システム
3 6 クライアント・コンピュータ・システム
3 8 ネットワーク・インタフェース・カード
4 0 ネットワーク・インタフェース・カード
4 2 ネットワーク・インタフェース・カード
4 4 ネットワーク・インタフェース・カード
1 2 1 キーボード
1 2 2 システム装置
1 2 3 主 CPU
1 2 4 システム RAM
1 2 5 システム ROM
1 2 6 大容量記憶装置
1 2 7 ディスプレイ
1 2 8 入出力論理回路
1 2 9 CD-ROM ドライブ
1 3 1 キーボード
1 3 2 ディスプレイ
1 3 3 主 CPU
1 3 4 システム・メモリ (RAM)
20 1 3 5 不揮発性メモリ (ROM)
1 3 7 ネットワーク・アダプタ・カード
2 0 0 ソフトウェア・コンポーネント
2 0 5 コントローラ
2 1 0 LAN 解析ソフトウェア
2 2 0 データ収集ソフトウェア
2 2 5 予測アルゴリズム
2 3 0 プッシュ・タスク
4 0 0 ホスト・システム
4 1 0 インターネットまたはイントラネット
30 4 1 2 クライアント・システム
4 1 4 クライアント・システム
4 1 6 クライアント・システム
4 2 0 J a v a 使用可能ブラウザ
4 2 2 キャッシュ

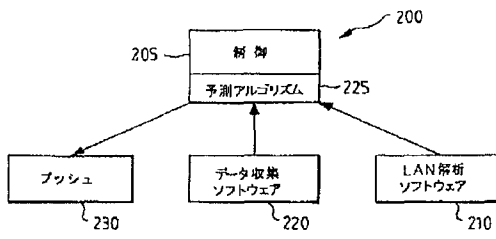
【図 1】



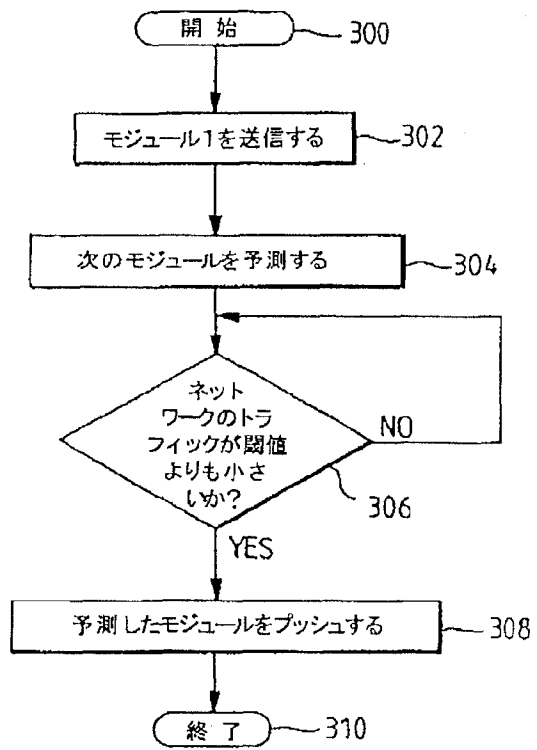
【図 2】



【図3】

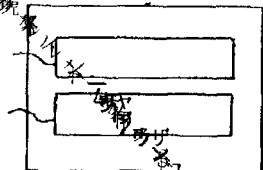


【図4】

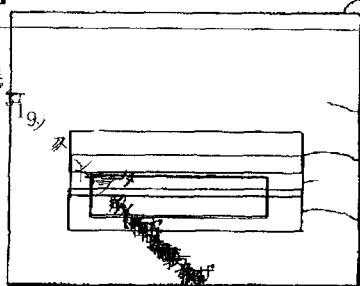


明査一

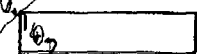
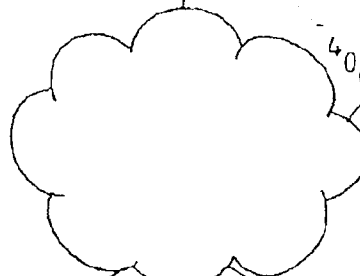
200



手



エ



開

1
2
3
2
1
1
5

